***Software Engineering***

***Software Requirements Specification***

***(SRS) Document***

**CareNet**

**February 22, 2023**

**V 1.0**

**By: Wesley Nguyen, Mecarah Son, Hasan Maanaki**

**We followed the UNCG Academic Integrity Policy.**

**Table of Contents**

[1.](#_heading=h.206ipza) Introduction 3

[1.1.](#_heading=h.kgcv8k) Purpose 3

[1.2.](#_heading=h.34g0dwd) Document Conventions 3

[1.3.](#_heading=h.1jlao46) Definitions, Acronyms, and Abbreviations 3

[1.4.](#_heading=h.43ky6rz) Intended Audience 4

[1.5.](#_heading=h.2iq8gzs) Project Scope 4

[1.6.](#_heading=h.xvir7l) Technology Challenges 4

[1.7.](#_heading=h.3hv69ve) References 4

[2.](#_heading=h.1x0gk37) General Description 4

[2.1.](#_heading=h.4h042r0) Product Perspective 4

[2.2.](#_heading=h.2w5ecyt) Product Features 4

[2.3.](#_heading=h.1baon6m) User Class and Characteristics 5

[2.4.](#_heading=h.3vac5uf) Operating Environment 5

[2.5.](#_heading=h.2afmg28) Constraints 5

[2.6.](#_heading=h.pkwqa1) Assumptions and Dependencies 5

[3.](#_heading=h.39kk8xu) Functional Requirements 5

[3.1.](#_heading=h.1opuj5n) Primary 5

[3.2.](#_heading=h.48pi1tg) Secondary 5

[4.](#_heading=h.2nusc19) Technical Requirements 6

[4.1.](#_heading=h.1302m92) Operating System and Compatibility 6

[4.2.](#_heading=h.3mzq4wv) Interface Requirements 6

[4.2.1.](#_heading=h.2250f4o) User Interfaces 6

[4.2.2.](#_heading=h.haapch) Hardware Interfaces 6

[4.2.3.](#_heading=h.319y80a) Communications Interfaces 6

[4.2.4.](#_heading=h.1gf8i83) Software Interfaces 6

[5.](#_heading=h.40ew0vw) Non-Functional Requirements 6

[5.1.](#_heading=h.2fk6b3p) Performance Requirements 6

[5.2.](#_heading=h.upglbi) Safety Requirements 7

[5.3.](#_heading=h.3ep43zb) Security Requirements 7

[5.4.](#_heading=h.1tuee74) Software Quality Attributes 7

[5.4.1.](#_heading=h.4du1wux) Availability 7

[5.4.2.](#_heading=h.2szc72q) Correctness 7

[5.4.3.](#_heading=h.184mhaj) Maintainability 7

[5.4.4.](#_heading=h.3s49zyc) Reusability 7

[5.4.5.](#_heading=h.279ka65) Portability 7

[5.5.](#_heading=h.meukdy) Process Requirements 7

[5.5.1.](#_heading=h.36ei31r) Development Process Used 7

[5.5.2.](#_heading=h.1ljsd9k) Time Constraints 7

[5.5.3.](#_heading=h.45jfvxd) Cost and Delivery Date 7

[5.6.](#_heading=h.2koq656) Other Requirements 7

[5.7.](#_heading=h.zu0gcz) Use-Case Model Diagram 8

[5.8.](#_heading=h.3jtnz0s) Use-Case Model Descriptions 8

[5.8.1.](#_heading=h.1yyy98l) Actor: Actor Name (Responsible Team Member) 8

[5.8.2.](#_heading=h.2y3w247) Actor: Actor Name (Responsible Team Member) 8

[5.8.3.](#_heading=h.1d96cc0) Actor: Actor Name (Responsible Team Member) 8

[5.9.](#_heading=h.3x8tuzt) Use-Case Model Scenarios 8

[5.9.1.](#_heading=h.2ce457m) Actor: Actor Name (Responsible Team Member) 8

[5.9.2.](#_heading=h.3bj1y38) Actor: Actor Name (Responsible Team Member) 9

[5.9.3.](#_heading=h.4anzqyu) Actor: Actor Name (Responsible Team Member) 9

[6.](#_heading=h.2pta16n) Design Documents 9

[6.1.](#_heading=h.1egqt2p) Software Architecture 9

[6.2.](#_heading=h.3oy7u29) High-Level Database Schema 9

[6.3.](#_heading=h.243i4a2) Software Design 9

[6.3.1.](#_heading=h.j8sehv) State Machine Diagram: Actor Name (Responsible Team Member) 9

[6.3.2.](#_heading=h.338fx5o) State Machine Diagram: Actor Name (Responsible Team Member) 9

[6.3.3.](#_heading=h.1idq7dh) State Machine Diagram: Actor Name (Responsible Team Member) 9

[6.4.](#_heading=h.42ddq1a) UML Class Diagram 9

[7.](#_heading=h.2hio093) Scenario 10

[7.1.](#_heading=h.wnyagw) Brief Written Scenario with Screenshots 10

# Introduction

## Purpose

[The goal of your project and the objectives it wishes to accomplish]

**The goal of CareNet is to allow patients to collaborate with their providers in order to take control of their care and coordinate with them about their health.**

## Document Conventions

[Full description of the main objectives of this document in the context of your project.

Here’s how you should begin this section:

“The purpose of this Software Requirements Document (SRD) is to...”

“In it, we will . . ., . . ., and . . ..”]

**The reason for the Software Requirements Document (SRD) is to outline the requirements for the CareNet patient portal management system. The SRD will describe both the client-view and developer-view requirements for the software. These requirements will focus on the functionality and user experience of the software, including how patients will interact with the application and the types of features they will have access to. Developer-oriented requirements will describe the system from a software developer's perspective, providing a detailed description of functional, data, performance, and other requisites. These requirements will focus on the technical aspects of the software, including the programming languages and tools that are used by the software architecture and any performance or security requirements that are fulfilled. By providing both client-view and developer-view requirements, this SRD will serve as a blueprint for the CareNet system, ensuring that it's developed to meet the needs of its intended users and stakeholders.**

## Definitions, Acronyms, and Abbreviations

[Include any specialized terminology dictated by the application area or the product area.

For example:]

| Java | A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager. |
| --- | --- |
| MySQL | Open-source relational database management system. |
| .HTML | Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content. |
| SpringBoot | An open-source Java-based framework used to create a micro Service. This will be used to create and run our application. |
| MVC | Model-View-Controller. This is the architectural pattern that will be used to implement our system. |
| Spring Web | Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system. |
| Thymeleaf | A modern server-side Java template engine for our web environment. This is one of the dependencies of our system. |
| NetBeans | An integrated development environment (IDE) for Java. This is where our system will be created. |
| API | Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage. |

## Intended Audience

[Describe which part of the SRS document is intended for which reader. Include a list of all stakeholders of the project, developers, project managers, and users for better clarity.]

**Patients: They serve as our main stakeholders and audience since they are the ones that benefit from the application the most. They have a direct interest in the quality of care they receive, including the cost and outcomes of their treatments.**

**Healthcare Providers: Providers such as doctors, nurses and other medical professionals are stakeholders because they provide the care that patients need. They are most concerned with patient outcome, job satisfaction and cost of delivering care.**

**Payers: Health insurance companies or government healthcare programs are essential stakeholders because they provide health coverage to their employees. They are interested in the quality of care, cost management, and ensuring that their members have access to the healthcare they need.**

## Project Scope

[Specify how the software goals align with the overall business goals and outline the benefits of the project to business.]

**The goal of this project is to give patients the ability to access all sorts of medical information in one place to provide a higher quality of care. This aligns with the goals of a healthcare organization, since they prioritize the needs of patients and their health.**

**The benefits of the project to business include:**

* **Relieving stress and pressure from healthcare providers and administrators as patients are given the opportunity to request for their medical information without the need for physical contact.**
* **Increasing pleasure to customers as they are given more power when they want to view their medical records in the comfort of their home.**
* **Reducing the amount of time that a patient needs to wait; therefore, increasing the amount of patients that are able to be served in hospitals within a day.**

## Technology Challenges

[Any technological constraints that the project will be under. Any new technologies you may need to use]

(Blank for V 1.0)

## References

[Mention books, articles, web sites, worksheets, people who are sources of information about the application domain, etc. Use proper and complete reference notation. Give links to documents as appropriate. You should use the APA Documentation model (Alred, 2003, p. 144).]

(Blank for V 1.0)

# General Description

## Product Perspective

[Describe the context and origin of the product.]

**The origin of CareNet involved leveraging the foundational framework of a pre-existing product, such as MyChart, and transforming it into a contemporary application that is designed to be more user-centric and readily available to its end-users.**

## Product Features

[A high-level summary of the functions the software would perform and the features to be included.]

**The app allows patients to access all their medical information in one place, reducing confusion and mistakes. It has features like scheduling appointments, viewing test results, refilling prescriptions, and more. The app is available online 24/7, so patients don't need to call their healthcare providers as often. CareNet helps patients take control of their health and helps healthcare providers provide better care.**

## User Class and Characteristics

[A categorization and profiling of the users the software is intended for and their classification into different user classes]

**Our website application does not expect our users to have any prior knowledge of a computer, apart from using a web browser. Our application will allow users to be able to access their medical history and patient profile without the need of physical contact with the provider.**

## Operating Environment

[Specification of the environment the software is being designed to operate in.]

**This application is designed for primarily desktop use.**

## Constraints

[Any limiting factors that would pose a challenge to the development of the software. These include both design as well as implementation constraints.]

(Blank for V 1.0)

## Assumptions and Dependencies

[A list of all assumptions that you have made regarding the software product and the environment along with any external dependencies which may affect the project]

The software will be dependent on Spring Web and Thymeleaf in order to create and execute the MVC architecture that will be developed within NetBeans. The application will also use the World Time API (http://worldtimeapi.org/) that will display the current date and time on the home dashboard for everyone to see.

# Functional Requirements

[Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.]

## Primary

[All the requirements within the system or sub-system in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements and constraints if any.]

* **FR0: The system will allow the user to make and reschedule appointments with their provider. The appointment will show up as a created appointment.**
* **FR1: The system will allow the user to message their provider and administrator with any queries. The system also lets the administrator/provider to message the patient.**
* **FR2: The system will enable the user to request prescription refills with their provider. The refill would need the provider to accept the request.**

## Secondary

[Some functions that are used to support the primary requirements.]

* **Authorization scheme so that patients can only see their portal and no other patients’ portal.**

# Technical Requirements

## Operating System and Compatibility

[The environments that will be needed to operate the system]

**The application will be compatible with any desktop operating system that is able to view and to interact with traditional web pages.**

## Interface Requirements

### User Interfaces

[The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.]

**CareNet should have the following functions for the application:**

* **The dashboard could include a menu bar or navigation panel that allows patients to access different sections of the app, such as their medical records, test results, prescription information, appointments, and messages from their healthcare provider.**
* **The user interface may also include visual representations of patient data, such as graphs or charts that display trends in a patient's health over time. It may have interactive features, such as the ability to request prescription refills, schedule appointments, or ask general medical questions.**
* **The interface may also have features that enable patients to communicate with their healthcare providers directly through the app, such as a messaging system or video chat functionality.**

### Hardware Interfaces

[All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.]

**The web application will run on any PC that has access to the internet, the ability to display webpages, and the ability to interact with web pages. This includes, but is not limited to desktop computers and laptops.**

### Communications Interfaces

[Determination of all the communication standards to be utilized by the software as a part of the project]

**It must be able to connect to the internet as well as the local database on mySQL.**

**The communication protocol, HTTP, must be able to connect to the HL7 return the EHRs, lab systems, and billing systems.**

### Software Interfaces

[The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them.]

**We will use React and Spring Boot ThymeLeaf to help build the frontend, as well as JPA for the backend database functionality. We will also use Spring Boot with Java to connect the frontend to the backend.**

# Non-Functional Requirements

[Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.]

## Performance Requirements

[The performance requirements need to be specified for all the functional requirements.]

* **NFR0(R): The patient will be able to make and reschedule appointments in less than 1 minute.**
* **NFR1(R): The patient will be able to send and receive messages in less than 1 minute.**
* **NFR2(R): The patient will be able to request a prescription refill in less than 1 minute.**

## Safety Requirements

[List out any safeguards that need to be incorporated as a measure against any possible harm the use of the software application may cause.]

* **Access controls: They should be implemented so that only authorized users can access patients’ health information, including user authentication, passwords policies, and privileges.**
* **Encryption: Data encryption should be used to protect the confidentiality of patient information when it is transferred and stored.**
* **Recovery plan: A back up should be implemented in the event that data loss occurs from malicious attacks.**

## Security Requirements

[Privacy and data protection regulations that need to be adhered to while designing of the product.]

* **NFR3(R): The system will only be usable by administrators, registered providers, and registered patients.**

## Software Quality Attributes

[Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.]

### Availability

The application must be available to use 24/7.

### Correctness

The application should be able to do any of the above requirements with little error.

### Maintainability

The software should be easy to maintain in case of any future changes.

### Reusability

Portions of code can be reused with changes to comply with the different user levels.

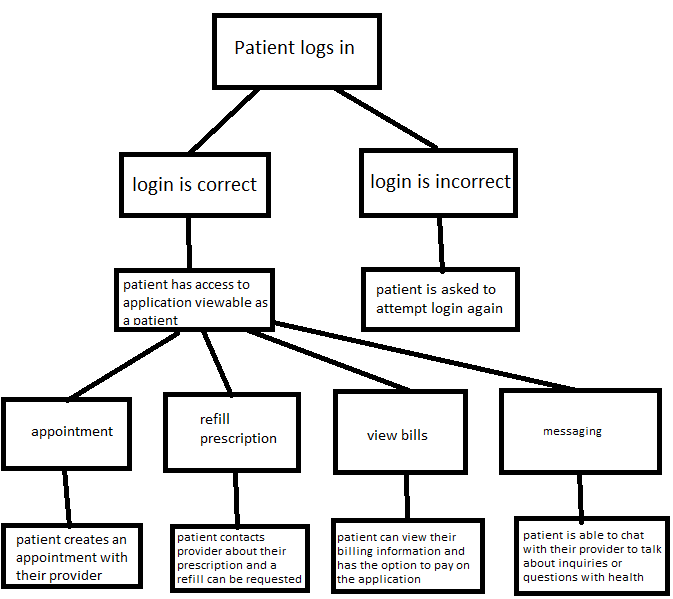
### Portability

The software should be portable between desktop computers and laptops.

## Process Requirements

### Development Process Used

[Software Process Model]



### Time Constraints

* **Regulatory Compliance: Our application is subject to regulatory requirements, such as HIPAA, that can add significant time and complexity to the development process.**
* **The development process of the application is dependent on the size of the development team and availability of resources, which could potentially slow down the progress made.**

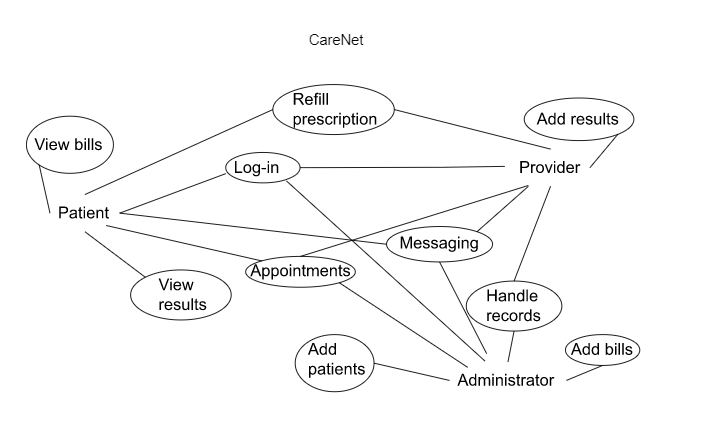
### Cost and Delivery Date

* **The cost of the application development will depend on the team working on the project and the complexity, so an estimate could not be determined at this stage.**
* **The delivery date of the project is also to be determined, since that it depends on several factors, such as regulation requirements and the resources needed, as well with unexpected delays.**

## Other Requirements

TBD

## Use-Case Model Diagram



## Use-Case Model Descriptions

### Actor: Patient (Mecarah Son)

* **Log-in**: Patient logs in to access medical records and other tools.
* **View results**: Allows the patient to view their test results.
* **View bills**: Allows the patient to view their own bills.
* **Refill prescription**: The patient can request a refill to their prescription
* **Messaging**: Send and receive messages to providers or administrators.
* **Appointments**: Patients can make appointments and reschedule them with their provider.

### Actor: Provider (Wesley Nguyen)

* **Log-in**: Provider logs in to access the hospital database.
* **Appointments**: Creates an updated work schedule so the appointment the patient makes is accounted for.
* **Messaging:** Notifies the client or administrator about any updates in regards to the patient or any other inquiries.
* **Refill Prescription:** Refills the prescription for the patient and notifies the patient.
* **Handle records:** Providers hold and record patient history and information and store them in the hospital database.

### Actor: Administrator (Hasan Maanaki)

* **Log-in**: Administrator logs in to view and access administrative tools.
* **Appointments**: Manage and reschedule appointments.
* **Messaging**: Receive and send notifications to providers or clients.
* **Handle records**: Administrator manages patient and provider information in the database.
* **Add patients**: Administrator adds new patients and manages patient access.
* **Add bills**: Administrator inputs invoices payable by patient for services rendered.

## Use-Case Model Scenarios

### Actor: Patient (Mecarah Son)

* **Log-in**:
  + **Initial Assumption**: The patient has a registered account that is saved in the database.
  + **Normal**: Patient enters username and password to login.
  + **What Can Go Wrong:** Patient’s credentials do not match the database credentials. The patient should be able to request a password reset.
  + **Other Activities**: The patient can reset their password through a link sent to their email.
  + **System State on Completion**: The patient is logged in and can access features of the app.
* **Refill prescription**:
  + **Initial Assumption**: The patient is logged in. The patient is able to request a refill for their prescription.
  + **Normal**: The patient sends a refill prescription request to their provider.
  + **What Can Go Wrong**: The patient is unable to request a refill and needs to message the provider.
  + **Other Activities**: The provider will have to allow the patient to request a refill.
  + **System State on Completion**: The request is sent to the provider successfully. The patient can see that their request is sent.
* **View bills**:
  + **Initial Assumption**: The patient is logged in. The patient is able to view their dashboard and see pending bills.
  + **Normal**: The patient can see their bills.
  + **What Can Go Wrong**: The bills are not shown or are incorrect due to an error.
  + **Other Activities**: The patient is allowed to see their past bills as well.
  + **System State on Completion**: The patient’s bills are shown when prompted.
* **Messaging**:
  + **Initial Assumption**: The patient is logged in. The patient is able to view their dashboard and go to messages.
  + **Normal**: The patient is able to communicate to providers and/or administrators.
  + **What Can Go Wrong**: The patient can message the incorrect recipient.
  + **Other Activities**: The patient can delete their message.
  + **System State on Completion**: Messages are sent and received successfully.
* **Appointments**:
  + **Initial Assumption**: The patient is logged in. The patient is in their appointments tab from the dashboard.
  + **Normal**: Appointments can be made by the patient to see their provider.
  + **What Can Go Wrong**: The patient makes an appointment at the wrong time or day.
  + **Other Activities**: The patient can cancel an appointment and reschedule.
  + **System State on Completion**: The appointment is made and the provider’s available appointments will be updated.
* **View results**:
  + **Initial Assumption**: The patient is logged in. The patient chooses to view results from the dashboard.
  + **Normal**: The patient is able to view their test results from their provider.
  + **What Can Go Wrong**: The patient is unable to view their test results.
  + **Other Activities**: The administrator/provider will fix the test results. The patient can refresh to view their test results.
  + **System State on Completion**: The test results are shown to the patient.

### Actor: Provider (Wesley)

* **Use-Case Name**: Login
  + **Initial Assumption**: The provider has a registered account that is used to sign into the hospital database.
  + **Normal**: The provider will input their credentials to log into the system.
  + **What Can Go Wrong**: The provider inputs the wrong information and is denied access to the system.
  + **Other Activities**: The provider will be able to select an option that will allow them to reset their passwords.
  + **System State on Completion**: The provider is able to enter into the system with their new credential.
* **Use-Case Name**: Appointment
  + **Initial Assumption**: The provider will see that the patient has made an appointment with them and the provider will create time to see them.
  + **Normal**: Appointments can be seen by the provider and are appropriately taken care of.
  + **What Can Go Wrong**: Providers may forget that they had an appointment with a patient that day and forget to see them or make room for them.
  + **Other Activities**: Providers may contact the patient and have a rescheduling with them.
  + **System State on Completion**: Provider goes to see patient at the new appointed time.
* **Use-Case Name**: Messaging
  + **Initial Assumption**: The provider would open the chat menu and talk with the patient and administrator.
  + **Normal**: The provider will be able to talk with patients and administrators about any inquiries that they would have.
  + **What Can Go Wrong**: Communication services could be temporarily down, causing lack of communication
  + **Other Activities**: Provider would email patient instead to get what they wanted to say across
  + **System State on Completion**: Patient receives the message from provider via email or by chat when it comes back
* **Use-Case Name**: Refill prescriptions
  + **Initial Assumption**: The provider issues prescriptions and refills for the patients.
  + **Normal**: The provider prescribes the patient with the proper medication.
  + **What Can Go Wrong**: The prescription is not legible or is incorrect and the patient receives the wrong medication.
  + **Other Activities**: The provider will receive the request by the patient for a replacement refill.
  + **System State on Completion**: The error is caught and the patient is issued the correct prescription.
* **Use-Case Name**: Add Results
  + **Initial Assumption**: After the visit, the provider uploads the results of the visit to the app for the patient to view.
  + **Normal**: The provider scans the documentations of the visit and uploads it to the hospital database.
  + **What Can Go Wrong**: There is incorrect information about the patient
  + **Other Activities**: The provider edits the mistakes and revise the paper for finalizing.
  + **System State on Completion**: The correct document is then uploaded to the database.

### Actor: Administrator (Hasan)

* **Use-Case Name**: Login
  + **Initial Assumption**: The administrator has a registered account with administrator privileges in the database.
  + **Normal**: Administrator inputs credentials to log into the system.
  + **What Can Go Wrong**: Administrators credentials are incorrect or they do not have administrator privileges.
  + **Other Activities**: Administrator password and privileges would both need to be manually changed in the database.
  + **System State on Completion**: The administrator can access the admin panel.
* **Use-Case Name**: Appointments
  + **Initial Assumption**: The administrator is logged in and the appointment panel is open.
  + **Normal**: All appointments are visible, and can be modified if needed.
  + **What Can Go Wrong**: Appointments could be scheduled during an unavailable time or overlapping appointments.
  + **Other Activities**: Administrator may contact provider regarding appointments if rescheduling is needed.
  + **System State on Completion**: Appointments are visible and scheduled correctly.
* **Use-Case Name**: Messaging
  + **Initial Assumption**: The administrator is logged in and the chat menu is open.
  + **Normal**: The administrator is able to message providers and patients.
  + **What Can Go Wrong**: Messaging is unavailable if communication services temporarily go down.
  + **Other Activities**: The administrator can send announcements that display on patients' frontpage.
  + **System State on Completion**: Messages can be received by and sent to patients and providers.
* **Use-Case Name**: Handle Records
  + **Initial Assumption**: The administrator is logged in and accessing a panel containing records.
  + **Normal**: The administrator can access and modify records.
  + **What Can Go Wrong**: Records could be accidentally deleted or modified incorrectly.
  + **Other Activities**: Deleted records can be restored.
  + **System State on Completion**: Records can be viewed and modified as needed.
* **Use-Case Name**: Add patients
  + **Initial Assumption**: The administrator is logged in and can view the provider-patient panel.
  + **Normal**: The administrator inputs patients details and selects providers.
  + **What Can Go Wrong**: Patient cannot be added without necessary information.
  + **Other Activities**: Can modify patients for providers and delete patients.
  + **System State on Completion**: Patients can be added or modified as needed.
* **Use-Case Name**: Add bills
  + **Initial Assumption**: The administrator is logged in and is accessing the billing panel
  + **Normal**: The administrator inputs invoices payable by patient for services rendered.
  + **What Can Go Wrong**: Billing errors such as incorrect amount owed, due to miscommunication.
  + **Other Activities**: Can view billing status for each patient.
  + **System State on Completion**: Invoices can be inputted and managed as needed.

# Design Documents

## Software Architecture

## High-Level Database Schema

## Software Design

### State Machine Diagram: Actor Name (Responsible Team Member)

### State Machine Diagram: Actor Name (Responsible Team Member)

### State Machine Diagram: Actor Name (Responsible Team Member)

## UML Class Diagram

# Scenario

## Brief Written Scenario with Screenshots